

Consistent Distributed Data

Konsolidierte logische Sichten auf physisch verteilte Daten

Werden Kopien von Daten physisch verteilt gespeichert, dann entstehen im Gesamtsystem inkonsistente Zustände. Consistent Distributed Data steht für Techniken wie Konsistenzprotokolle oder Conflict-Free Replicated Data Types (CRDT), die trotz aller Latenzen eine konsistente Außensicht auf diese Daten sicherstellen sollen.

Definition

Ein System mit verteilten Datenkopien kann vorübergehend in Zustände inkonsistenter Kopien der Daten verfallen, etwa weil eine Änderung noch nicht oder noch nicht vollständig disseminiert wurde. Consistent Distributed Data verarbeitet, speichert und verwaltet die Daten und ihre Kopien so, dass aufrufende Systeme oder Anwender vor Inkonsistenzen geschützt sind und das Gesamtsystem möglichst einheitlich wahrnehmen können. Zwei mögliche Wege stehen dabei zur Auswahl: **Konsistenzprotokolle** und **Conflict-Free Replicated Data Types**.

Ein Konsistenzprotokoll schreibt Abfolgen von Nachrichten innerhalb eines Clusters vor. Diese Nachrichten sollen sicherstellen, dass jeder Knoten weiß, ob eine gegebene Information möglicherweise nur lokal vorhanden ist oder bereits für das Gesamtsystem gilt. Die Knoten eines Clusters handeln dazu unter sich auf Basis festgelegter Protokollschritte einen zufälligen **Leader** aus. Die übrigen Knoten fungieren danach als **Follower**. Jeder Follower kann zu jeder Zeit den Leader benennen. Während sich die Daten bei einem beliebigen Knoten lesen lassen, können sie ausschließlich beim Leader geschrieben werden. Der Leader sammelt die Schreibanfragen der Klienten ein und verteilt sie geordnet, in periodischen Nachrichten an alle Follower. Die Follower antworten wiederum mit Bestätigungen. Eine Änderung, die

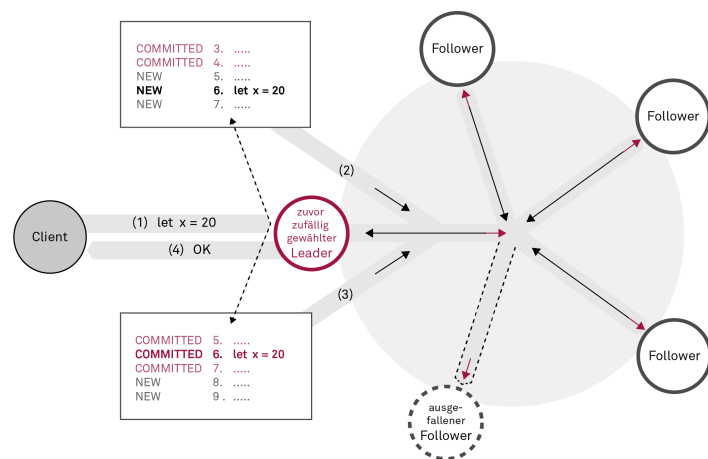
durch mehr als die Hälfte der Follower bestätigt wurde, wird in der nächsten periodischen Nachricht des Leaders als **committed** gemeldet. Die Tatsache, dass nicht alle Knoten bestätigt haben müssen, bedeutet, dass der Ausfall einzelner Knoten zu keinem Gesamtausfall des Clusters führt.

Im Gegenzug dazu lösen **Conflict-Free Replicated Data Types**, kurz CRDT, das Problem auf der Ebene der Daten selbst. Eine CRDT-Instanz sammelt neben den reinen Nutzdaten deshalb auch Metadaten, die detaillierte Informationen über die Änderungshistorie auf dem jeweiligen Knoten beinhalten. Änderungen lassen sich so an andere Knoten verteilen und dort ebenfalls ausführen. In Konfliktfällen greifen Exekutionsregeln, die fachlich sogar willkürlich erscheinen können, etwa wenn der Knoten mit der niedrigsten zufällig zugewiesenen laufenden

Nummer die Priorität erhält. Weil die Regeln aber für das Gesamtsystem gelten, stellen sie sicher, dass sämtliche Knoten nach Durchführung sämtlicher Änderungen dieselbe Sicht auf die Nutzdaten haben.

Referenzszenario

Die gleichen Daten sollen an mehreren Stellen gleichzeitig und gleichberechtigt bearbeitet werden können, etwa von verschiedenen Mitarbeitern und Abteilungen, an mehreren Standorten eines Unternehmens oder zwischen mehreren Unternehmen. Weil alle Teilnehmer ihre Datenkopie gleichberechtigt bearbeiten dürfen, sind Konflikte unausweichlich. Nicht-funktionale Anforderungen machen es nötig, diese auftretenden Konflikte der unterschiedlichen Datenstände automatisch aufzulösen, um eine für alle Teilnehmer einheitliche Sicht auf die Daten zu gewährleisten.



Nicht-funktionale Anforderungen

- Konsistenz
- Ausfallsicherheit
- Skalierbarkeit
- ggf. Robustheit gegenüber Betrug

Technologiekontexte

- RDBMS
- NoSQL
- Blockchain
- Clusterverwaltung

CDD

Speicherungsarchitekturen

- Master-Slave
- Master-Master
- Replikationsstrategie
- Consistency Level

Verwandte Architekturtrends

- Asynchronous Programming
- Event-Driven Architectures
- Microservices
- Command Query Responsibility Segregation

Potenzial

Der zunehmende Bedarf nach gleichzeitiger, verteilter Arbeit erhöht auch den Bedarf nach konsistenten Datenkopien, etwa wenn Mitarbeiter unterwegs vorübergehend vom Internet getrennt sind oder schlicht mehrere Mitarbeiter gleichzeitig an Daten arbeiten können sollen. Insbesondere im Umfeld von Unternehmenslösungen oder bei Service-Anbietern ist dies der Fall. Consistent Distributed Data lässt sich in solchen Fällen im Grunde als unverzichtbar und als einzig gangbare Lösung bezeichnen.

Der Verzicht auf Consistent Distributed Data kann einem Unternehmen wiederum als rückständig und innovationsarm ausgelegt werden. Schließlich erwarten Anwender eine ebensolche Funktionalität bereits von Software-Produkten.

Reifegrad

Konzeptionell ist das Thema durchdrungen, verstanden und auch über die vielen Jahre des Einsatzes verteilter Systeme ausgereift. Dennoch gibt es unter den vielen Frameworks – mit subtil unterschiedlichen Vorgehensweisen – einige Vertreter, die softwaretechnisch unausgegoren sind. Dies ist insbesondere bei den CRDTs der Fall. Einem produktiven Einsatz steht aber nichts im Wege.

Marktübersicht

Auf der konzeptionellen Seite ist **Paxos** als erstes Konsistenzprotokoll bereits seit 1989 im Einsatz. Aufgrund einiger Unzulänglichkeiten bekommt in modernen Systemen aber zunehmend das 2013 entworfene, vollständigere und leichter verständliche **Raft** den Vorzug gegenüber Paxos.

Für den Produktiveinsatz aus Unternehmenssicht bereitstehende Bausteine sind Consul und Etcd, CockroachDB sowie Atom Collaborative Editing. Sie dienen als Grundlage für individuelle, fachliche Softwarelösungen. Ferner stellen viele Service-Anbieter bereits fertige Lösungen für verteiltes, kollaboratives Arbeiten bereit.

Alternativen

Consistent Distributed Data setzt vor allem auf die automatisierte Konsensbildung. Eine Alternative ist daher der klassische, manuelle Abgleich der Datenbestände, der langwierig, kostenintensiv und in hohem Maße fehleranfällig ist.

Fazit

- + skaliert bei Datenverteilung und Datenzugriff mit herkömmlicher Hardware und bei hoher Latenz
- + bietet auf fachliche Probleme spezialisierte Frameworks
- + ist meistens als frei verfügbare und quelloffene Lösung verfügbar
- erfordert produktspezifisches Expertenwissen bei Wahl und Konfiguration der passenden Lösung
- kann Anwender mit überraschenden Eingeständnissen konfrontieren
- offenbart nicht-deterministische Fehler erst bei aufwändigen Lasttests oder in Produktion



Buzzword Factor (Ent./Customer)

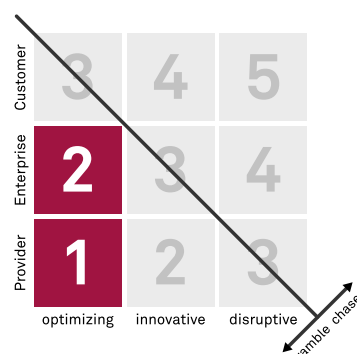
1 low	2 medium	3 high
-------	----------	--------

Entry Barrier (Provider)

1 low	2 medium	3 high
-------	----------	--------

Benefit Level (Provider)

1 low	2 medium	3 high
-------	----------	--------



<https://msg.direct/techrefresh>

Stand: September 2020

msg systems ag

Robert-Bürkle-Straße 1 | 85737 Ismaning/München | Telefon: +49 89 96101-0 | Fax: +49 89 96101-1113 | www.msg.group | info@msg.group